

- 37 -

We Claim:

1. A method of type-checking a code segment written in a programming language comprising:

5 translating the code segment from the programming language to one or more representations of an intermediate language; and

type-checking the one or more representations based on a rule set, wherein the rule set comprises rules for type-checking a type that indicates an element of the representation can be one of a plurality of types.

10

2. The method of claim 1 wherein the type that indicates an element of the representation can be one of a plurality of types is an unknown type.

3. The method of claim 1 wherein the rule set is selected from a plurality
15 of rule sets.

4. The method of claim 3 wherein only a fraction of the rule sets contain rules for type-checking a type that indicates an element of the representation can be one of a plurality of types.

20

5. The method of claim 1 wherein the rule set further comprises rules for type-checking types representing categories of types found in a plurality of programming languages.

25 6. A method of selectively retaining type information during compilation in a code segment written in a programming language, the method comprising:

- 38 -

translating the code segment from the programming language to one or more representations of an intermediate language;

for each representation, determining whether to retain type information for one or more elements of the representation; and

5 based on the determination, associating one or more elements of the representation with a type indicating the element can be of any type.

7. The method of claim 6 wherein the determination is based on a current stage of compilation, a characteristic of each representation, or the programming
10 language.

8. The method of claim 6 further comprising a rule set for type-checking the one or more representations, wherein the rule set comprises rules for type-checking the type indicating the element can be of any type.
15

9. The method of claim 6 wherein the type indicating the element can be of any type has size information associated with it.

10. The method of claim 9 further comprising generating code from at least
20 elements associated with the type indicating the element can be of any type based on the size information.

11. The method of claim 6 wherein the type indicating the element can be of any type is an unknown type.
25

12. A method of translating types associated with a plurality of programming languages to types of an intermediate language, the method comprising:

- 39 -

replacing the types associated with the plurality of programming languages with the types of the intermediate language, wherein the types of the intermediate language comprise general categories of the types associated with the plurality of programming languages and an unknown type.

5

13. The method of claim 12 wherein the types of the intermediate language further comprise types related to programming language specific primitive types.

14. A system for type-checking an intermediate representation of source
10 code in a compiler comprising:

one or more types associated with elements of the intermediate representation, wherein at least one of the types indicates an element can be one of a plurality of types;

one or more rule sets comprising rules associated with a type indicating
15 an element can be one of a plurality of types; and

a type-checker for applying the one or more rule sets to the elements of the intermediate representation.

15. The system of claim 14 wherein the type indicating the element can be
20 one of a plurality of types has size information associated with it.

16. The system of claim 14 wherein the type indicating an element can be one of a plurality of types is an unknown type.

25 17. The system of claim 14 wherein the one or more rule sets applied to the elements of the intermediate representation are selected based on the stage of compilation.

- 40 -

18. The system of claim 14 wherein the one or more rule sets applied to the elements of the intermediate representation are selected based on a characteristic of the source code.

5 19. The system of claim 14 wherein the one or more rule sets applied to the elements of the intermediate representation are selected based on the intermediate representation.

20. The system of claim 14 wherein only a fraction of the one or more rule
10 sets contain rules for type-checking a type that indicates an element can be one of a plurality of types.

21. The system of claim 14 wherein the one or more rule sets further
comprise rules for type-checking types representing categories of types found in a
15 plurality of programming languages.

22. The system of claim 14 wherein the system selectively retains type
information for some elements of the intermediate representation and selectively does
not retain type information for other elements of the intermediate representation.
20

23. The system of claim 22 wherein the system selectively does not retain
type information for an element of the intermediate representation by replacing a type
associated with the element with the type indicating an element can be one of a
plurality of types.
25

- 41 -

24. A method of representing types in an intermediate language comprising:

defining a plurality of types to be associated with elements of the intermediate language, wherein one of the plurality of types indicates that an element of the
5 intermediate language is associated with an unknown type.

25. The method of claim 24 wherein the type indicating that an element of the intermediate language is associated with an unknown type has a size associated with it.

10

26. The method of claim 24 wherein an element of the intermediate language that was previously associated with another type is associated with the type indicating that the element is associated with an unknown type.

15 27. The method of claim 24 wherein the plurality of types further comprises types representing categories of types found in a plurality of programming languages.

28. A computer-readable medium containing computer-executable
20 instructions for implementing the method of claim 24.

29. A computer-readable medium containing computer-executable instructions for implementing the method of claim 1.

25